Sequence-State Coding for Digital Transmission

By P. A. FRANASZEK

(Manuscript received May 9, 1967)

A systematic approach to the analysis and construction of channel codes for digital baseband transmission is presented. The structure of the codes is dominated by the set of requirements imposed by channel characteristics and system operation. These requirements may be translated into symbol sequence properties which, in turn, specify a set of permissible sequence states. State-dependent coding of both fixed and variable length is a direct result. Properties of such codes are discussed and two examples are presented.

I. INTRODUCTION

1.1 *General*

Binary information to be transmitted over a baseband digital system must typically be encoded into a sequence of symbols suitable for transmission through the channel. The structure of such codes is dominated by the set of requirements imposed by considerations such as channel characteristics and system operation. Several codes designed for baseband transmission have been discussed in the literature,[1, 2, 3] but the analysis of such coding has received little attention. This paper presents a systematic approach to the description and construction of codes for such application. The symbol sequence to be generated by the code is viewed as the output of a sequential machine with a set of permissible states derived from the imposed requirements. State-dependent codes, both of fixed and of variable length, are a direct result. Various properties of such codes are discussed, and two examples are presented.

An advantage of using a sequence-state point of view is the ease with which it may be adapted to computer analyses, which are feasible due to the typically short word length of such codes (usually less than eight symbols), and useful because normal design procedure often entails the construction of a number of codes, derived from different sequence requirements, in order to compare properties.

1.2 *System Requirements*

A block diagram of a baseband digital transmission system is shown in Fig. 1. Binary signals are encoded into a sequence of pulses (symbols) which are transmitted over a channel with regularly spaced reconstructive repeaters.[4] Typically, a number of factors strongly influence the choice of a channel code. Examples are:

(*i*) The symbol sequence is often required to furnish timing information to the repeaters.

(*ii*) The channel may impose restrictions on the spectral shape of the sequence. Most such systems, for example, have a transmission null at dc. This implies that long strings of symbols of one polarity cannot be tolerated.

(*iii*) Provisions for monitoring the channel error rate during normal operation may be necessary.

(*iv*) Pulse sequence requirements must be satisfied independently of the source statistics.

Requirements such as the above may usually be translated into limitations on the length of strings of like symbols, specifications of sets of allowable transitions between symbols (such as when a $+2$ pulse must be followed by a -2 pulse to avoid timing jitter due to asymmetries), and bounds on the variation of the running digital sum. The latter is defined as

$$\xi(T) = \sum_{t=T_0}^{T} a_t ,$$

where $\{a_t\}$ are the weights assigned to the channel symbols and T_0 is an arbitrary but fixed time.

The objective of the code designer is to specify a code which meets the imposed conditions while obtaining maximum information capacity (average number of bits per channel symbol) or alternatively to optimize sequence properties.

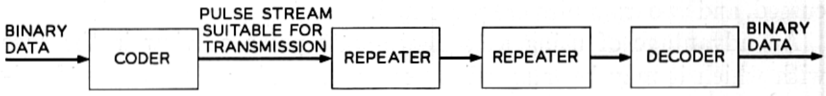

Fig. 1 — Block diagram of a baseband digital transmission system.

1.3 *Permissible Sequences*

A code may be defined as a mapping from the binary input information to the ensemble of symbol sequences which satisfy the imposed criteria. The allowable symbol sequences can, in turn, be specified by a set of permissible states $\{s_i\}$ for each point, describing such quantities as the present value of the running digital sum, the preceding number of symbols of a given type, etc. In addition to the $\{s_i\}$, it is convenient to define a set of allowable words $\{w_j\}$, which take the sequence through a succession of allowable states, one for each symbol in the word. A necessary condition for w_j to be permissible is that there exists at least one state s_r such that the states resulting from the use of w_j at s_r are all allowable. In addition, the words to be used in the code may be restricted to a subset of $\{w_j\}$ which satisfies requirements not specified by a particular choice of states, such as various error-correcting properties. The latter restriction will not be treated in this paper.

It is assumed in this paper that each code word carries an integral number of information bits, and that the ratio of bits per symbol remains constant over each word. The latter assumption is a sufficient condition for synchronous transmission.

II. FIXED LENGTH CODE STRUCTURE

2.1 *General*

The simplest codes of the above type are those of fixed length, which may be defined as mappings from the set of input binary words of length $\log_2 L$ to the set of allowable words $\{w_j\}$ of length N. These mappings are generally state-dependent. That is, the choice of a code word may be a function of the state occupied by the sequence.

It is convenient to introduce a number of definitions.

Definition: $W(s_i)$ is the set of words (alphabet) of length N which take the symbol sequence from state s_i through a succession of allowable states.

The fact that each word w_j must carry $\alpha = \log_2 L$ information bits implies that words used in a code must terminate in states for which $W(s)$ contains a minimum of 2^α words.

Definition: The principal states $\{S_m\}$ of the channel sequence are those states for which $W\{S_m\}$ contains at least 2^α words each of which takes the sequence to another principal state.

The existence of a set of principal states is a necessary and sufficient condition for the existence of a code. It is possible to implement computer search routines for finding such states. The appendix contains a description of a search method which was used as an aid on constructing the two codes in Section 5.2.

A given code actually occupies a subset $\{\sigma_n\}$ of the principal states at the end of code words. Let $W^*(\sigma_n) \subset W(\sigma_n)$ be the words actually used for coding when the sequence occupies the state σ_n.

Definition: σ_n and σ_k are in the same *alphabet class* R_q if and only if $W^*(\sigma_n) = W^*(\sigma_k)$.

The above relation partitions the set of terminal states into alphabet classes R_q, $q = 1, 2, \ldots Q$. A set of words $W^*(R_q)$ is associated with each such class. Each $W^*(R_q)$ contain 2^α words which are used when the sequence occupies one of the terminal states in the class R_q (i.e., each word in $W^*(R_q)$, $q = 1, 2, \ldots Q$ is associated with one binary input word b_e). It is advantageous to minimize the number of alphabet classes in order to simplify the coding and decoding circuits.

The coder tracks the state of the symbol sequence, and at the end of each word, codes the next binary word b_e into a word chosen from the alphabet corresponding to the state occupied by the output sequence. For example, the Paired Selected Ternary[1] code book consists of two alphabets, which contain words of zero or positive weight and zero or negative weight, respectively. The coder switches to a different alphabet after use of a word with non-zero weight.

2.2 *State-Independent Decoding*

The class of codes defined in Section 2.1 may be uniquely decoded provided the state of the symbol sequence is known. Frequently, however, it is not feasible for the decoder to distinguish which member of a given subset of states the sequence occupies at the end of a code word. For example, the states may be functions of the instantaneous symbol sequence sum. Here a single error in detection may result in an unbounded string of decoding errors. In addition, implementation of state-dependent decoding requires circuits to track the sequence state. Thus, state-independent decoding may often be necessary.

Decoding independently of the state is possible if and only if one binary word b_e corresponds to each code word $w_i \in \{w_j\}$. That is, the state-dependent mapping of $\{b_e\}$ into $\{w_j\}$ must have a unique inverse. If decoding is to be independent of the state within a given subset of the terminal states, the above mapping must have a unique inverse within this subset.

Given a group of terminal states $\{\sigma_q\}$, $q = 1, 2, \ldots Q$, and their associated alphabets $W^*(\sigma_q)$ [each containing 2^α words], it is desirable to determine whether it is possible to code so that decoding is independent of q. The following are necessary and sufficient conditions for the above:

Condition 1: A sufficient, but not necessary, condition for the existence of an assignment of binary words to the members of $W^*(\sigma_t)$ $q = 1, 2, \cdots Q$ so that decoding is independent of t is that for any integers u, v such that $1 \leqq u < v \leqq Q$

$$w_r \varepsilon W^*(\sigma_u) \cap W^*(\sigma_v) \rightarrow w_r \varepsilon W^*(\sigma_{u+1}).$$

Proof: Suppose that a binary word b_e has been assigned to w_r in the alphabet $W^*(\sigma_u)$. Then it is possible to give w_r the same binary assignment in $W^*(\sigma_{u+1}) \cdots W^*(\sigma_v)$. The lack of necessity of the above condition may be demonstrated by a renumbering of the alphabets.

Condition 2: A necessary and sufficient condition for the possibility of assigning binary words to the members of $W^*(\sigma_\theta)$, $\theta = 1, 2 \cdots \Theta$ so that decoding is independent of θ is that for $1 \leqq u \leqq v \leqq x \leqq \Theta$ and for each

$$w_r \varepsilon W^*(\sigma_u) \cap W^*(\sigma_x)$$

such that

$$w_r \notin W^*(\sigma_v)$$

there exists a $w_p \varepsilon W^*(\sigma_v)$ such that there exists no σ_m, $1 \leqq m \leqq \Theta$, for which w_r, $w_p \varepsilon W^*(\sigma_m)$.

Proof: Sufficiency follows from condition 1 by requiring that w_p and w_r be equivalent (in the sense that they are assigned the same binary word).

The necessity of the condition follows from the fact that if

$$w_r \notin W^*(\sigma_v)$$

then some other word $w_p \varepsilon W^*(\sigma_v)$ must be assigned the binary word assigned to w_r in the uth alphabet. But if w_p, $w_r \varepsilon W^*(\sigma_m)$, $1 \leqq m \leqq T$, state-independent decoding is precluded.

It is obvious from the above conditions that state-independent decoding is always possible when there are only two alphabets (as in Paired Selected Ternary[1]). An example of a group of alphabets which do not admit state-independent decoding is shown in Table I.

2.3 *Framing*

The received symbol sequence must be correctly partitioned into blocks of words length N before decoding. This may be done by observing whether the received words coincide with the words utilized in a code, by tracking the sequence state to determine whether ends of the received words coincide with sequence terminal states, or by looking for the presence of nonallowable sequences of permissible words.

III. VARIABLE LENGTH CODES

3.1 *General*

Attempts to increase fixed length state-dependent code efficiency with constraints on pulse sequence properties result in increased word length, and thus in rapidly mounting coder and decoder complexity, reframe times and sensitivity to detection errors. Hence, variable length codes, which may combine the advantages of short and long word lengths, are frequently advantageous.

Consider a sequence of symbols generated by a fixed length code of word length N. The sequence by definition occupies a principal state at the end of each word. Often, however, a principal state will be entered in the middle of a word. This offers the possibility of using the word fragment to convey information, and of starting any of at least 2^α words from this point (by definition of a principal state). The result, subject to a number of restrictions outlined below, is a variable length code.

Variable length codes offer the possibility of using short words more frequently than those of longer lengths. This often permits a marked decrease in coder and decoder complexity relative to a fixed length code of like efficiency and sequence properties, since the number of stored words may be far smaller. For example, VL43 (described below), a three alphabet variable length code, uses a total of 56 words,

TABLE I—A SET OF ALPHABETS WHICH DO NOT ADMIT STATE-INDEPENDENT DECODING.

$W^*(\sigma_1)$	$W^*(\sigma_2)$	$W^*(\sigma_3)$	$W^*(\sigma_4)$
w_1	w_1		
w_2		w_2	
w_3			w_3
		w_4	w_4
w_5	w_5	w_5	w_5
	w_6	w_6	w_6
	w_7		

binary words: b_1, b_2, b_3, b_4

compared to the 256 required for a single alphabet of a fixed length code of equal efficiency and sequence properties.

3.2 *Variable Length Code Structure*

The structure of variable length codes required to satisfy sets of sequence requirements is quite similar to that of fixed length codes. A number of special features, however, arise from the presence of words of different lengths.

The requirement of synchronous transmission, coupled with the asumption that each word carries an integer number of information bits, implies that the word lengths in a given code are integer multiples of a *basic word length* N, where N is the smallest integer for which the bit per symbol ratio α/N is that of two integers (i.e., if the bit per symbol ratio is 1.5, the basic word length is 2). The proof of this statement follows from the fact that the word length K_iN must be such that the number of bits per word $K_i\alpha$ is an integer [this paper covers the transmission of binary data]. Suppose that $K_i = n + b$, where n is an integer and $0 < b < 1$. Then $b\alpha$ and bN must also be integers, which violates the definition of N.

The set of allowable words may be written as $\{w_{ji}\}$, $i = 1, 2, \ldots,$ M, where w_{ij} is the jth word of length iN.

Definition: $W_i(s_l)$ is the set of words of length iN, $i = 1, 2, \ldots, M$, which may be used in the permissible state s_l without violating the sequence requirements.

$$W(s_l) = \bigcup_{i=1}^{M} W_i(s_l).$$

A set of principal states may be defined for a symbol sequence associated with a variable length code in a manner analogous to that of Section II.

Definition: The principal states $\{S_l\}$ are those allowable states for which $W(S_l)$ contains sufficient words terminating in principal states to maintain an information rate of α bits per N symbols.

The number of words of length iN, $i = 1, 2 \ldots M$ required to maintain a bit rate of α bits per N symbols is given by

$$L_1[W(s_l)] + L_2[W(s_l)]2^{-\alpha} + \cdots + L_M[W(s_l)]2^{-(M-1)\alpha} = 2^{\alpha},$$

where $L_i[W(s_l)]$ is the number of words in $W_i(s_l)$. Equation (1) clearly indicates the advantage of using as many short words as possible.

The terminal states $\{\sigma_m\}$ are as defined in Section II. Similarly, $W^*(\sigma_m)$ and $W^*(\sigma_m)$ are those members of $W_i(\sigma_m)$ and $W(\sigma_m)$, respectively, which are actually used in the code; two terminal states σ_m and σ_p are in the same alphabet class if and only if $W^*(\sigma_m) = W^*(\sigma_p)$.

It is clear that

$$w_{ji} \neq P_i(w_{kr})$$

for all w_{ji} , w_{kr} ε $W^*(\sigma_e)$ and $w_{ji} \neq w_{kr}$. $P_i(w_{kr})$ denotes the first iN symbols of the word w_{kr} . The above condition follows from the fact that w_{ji} ε $W(S_e)$ implies that w_{ji} , when used in the state S_e , takes the sequence to another principal state.

3.3 *State-Independent Decoding*

State-independent decoding of a variable-length code is desirable for the same reasons as for one of fixed length. The criterion for such decoding is the same as for a fixed length code — the state-dependent mapping from the binary words $\{b_e\}$ to the words to be transmitted in the channel must have a unique inverse. Two considerations which affect the complexity of this mapping are the presence of words of length iN, $i = 1, 2, \ldots, M$ and the existence of words which consist of concatenations of two or more shorter words. Such word combinations will be called *concatenated* words. It follows from Section 3.2 that a concatenated word and its prefix cannot occupy the same alphabet. From Section 3.2, it can be concluded that a concatenated word of length iN is a legitimate word in a given alphabet if no terminal state is reached with the KNth symbol for $0 < K < i$. A requirement for state-independent decoding is that the binary words assigned to a concatenated word be concatenations of the binary words assigned to the words of which the concatenated word is composed. Section 5.2 contains an example of a variable-length code which satisfies these criteria.

3.4 *Framing*

The process of framing (at the decoder) a symbol sequence generated by a variable-length code may be divided into two parts: partitioning the sequence into blocks of length N (the basic word length), and properly grouping these blocks into words of length iN, $i = 1, 2, \ldots, M$. Framing may be performed in much the same manner as for a fixed-length code. A misframe condition may be detected by noting the presence of nonpermissible words, word sequences which do not occur in the code, and/or the occupation of a nonterminal state at the end of a word.

It is often possible to substantially simplify the framing process by ensuring that secondary misframing (the condition that results from improper grouping of blocks of length N into words) is self-terminating. This restricts the need for framing to blocks of length N (the basic word length), and frequently results in superior framing performance compared to a fixed length code of equal efficiency and sequence properties.

Consider a state-independent decoder which stores all word prefixes of length iN, $i = 1, 2, \ldots, M-1$, and which operates in the following manner: If the first fN symbols that are received form a prefix of a legitimate word, the decoder considers the first $(f+1)\ N$ symbols. Otherwise the first fN symbols are decoded into an arbitrary binary word of length $f\alpha$. The concatenated words are treated as strings of the shorter words of which they are composed.

A sufficient condition for secondary misframing to be self-terminating in the above decoder is that (excluding concatenated words)

$$E_r[w_{ki}] \neq P_r[w_{me}]$$

for all r, k, i, m, e such that $e > r$ and $i > r$. $E_r[w_{ki}]$ denotes the last rN symbols of the word w_{ki}. This ensures that the secondary misframing condition terminates with the ending of the word w_{ki}.

IV. PERFORMANCE MONITORING

It is often necessary to monitor the level of channel performance without interrupting service. Codes with bounded sequence sums readily lend themselves to this objective, especially in the low noise environment typical of telephone communications systems. This is because an error is quite likely to eventually cause the sequence sum to exceed the imposed bounds. Alternatively, the behavior of the digital sequence at the bound can be observed. For example, not more than one consecutive zero (null pulse) might be permitted at extreme sequence sum values, while strings of length $N > 1$ may occur at intermediate sums. Here, the detection of a string of zeros at an extreme sequence sum value indicates that an error has occurred in detection. An advantage of these methods[1] is that error monitoring can be performed without framing.

Standard methods of error monitoring (or error detection) can also be incorporated in a code of this form. A possibility is the requirement that the words have even digital sums, or more generally that the terminal states $\{\sigma_j\}$ be separated by a minimum "distance."

V. CODE CONSTRUCTION

The foregoing sections have described a number of features of state-dependent codes. Steps in the construction of a code in this class are outlined below.

(*i*) Specification of the required channel symbol sequence properties.

(*ii*) Translation of the requirements into a set of acceptable states.

(*iii*) Search for the permissible words in each allowable state. Search for the principal states. The above can be done either by hand or by a computer search routine such as that described in the appendix.

(*iv*) Terminal states and code words are chosen for (if necessary) state-independent decoding, to optimize spectral properties, error monitoring and framing statistics. This usually requires much tedious computation.

It may occur that a given set of sequence requirements are such that a set of principal states does not exist for codes of word length N, but that a length of MN yields an acceptable code. In this case, a variable length code of basic word length N with maximums length MN might be a possibility. It is advantageous to use as many words of length N as possible, and to utilize the longer lengths to achieve a sufficient number of words inn each terminal state.

VI. EXAMPLES

The following two ternary codes, one of fixed length and one of variable length, are examples of results of the above procedures. These codes were designed for possible use in a future high-speed baseband digital communication system.[5] The symbols to be transmitted on the line consist of positive, negative and null pulses. The main objectives in the design of these codes was to restrict the variations in the running digital sum of the pulse stream (with positive, negative, and null pulses assigned values of $+1$, -1, and 0 respectively) to avoid dc buildup in the channel, and to have as many transitions between symbols as possible in order to provide energy for self-timing in the repeaters.[4]

The information capacity of each code is four bits for every three symbols. The allowable states are described in terms of the running digital sum. Words were chosen from those permissible in each state so as to maximize the number of transitions between symbols. The search routine described in the appendix was used as an aid in code construction.

6.1 *A Fixed Length Selected Ternary Code*

The MS43 selected ternary code (Table II) is an example of a fixed length, three alphabet code of word length three. The digital sequence sum varies over six levels, which are arbitrarily labeled zero to five. These six levels are the allowable states. Terminal states occur only at values of one to four inclusive. There are three alphabet classes. R_1 and R_3 correspond to terminal states with sequence sums of one and four respectively, while R_2 corresponds to a sum of either two or three. The choice of MS43 words is such that strings of zeros and symbols of like sign are limited to lengths of four and five respectively.

Coding is performed as a function of the terminal state. For example, if 00110100 is to be coded when the value of the sequence sum is $+1$, the first four bits are coded as $0-+$ and the second as $0++$. It can easily be seen from Table II that decoding is independent of the state.

6.2 *A Variable Length Selected Ternary Code*

The VL43 code [Table III] is an example of a three alphabet selected ternary code of variable length. The basic word length is three and the maximum length six. The bit per symbol ratio is 4/3, as in MS43. Secondary misframing is self-terminating and decoding is state-independent. The variable length feature permits a decrease in the variation of the running digital sum from six to five levels, and a significant increase in density of transitions. There are three terminal states (σ_1, σ_2, σ_3) each with its own alphabet. $W^*(\sigma_2)$ contains only words of length three. $W^*(\sigma_1)$ and $W^*(\sigma_3)$ each contain 16 words of length six (each carrying eight bits) and 15 words of length three. Both $W^*(\sigma_1)$ and $W^*(\sigma_3)$ contain concatenated words.

Coding is performed in much the same manner as for MS43. For example, if 110000010000 is to be coded when the state is σ_2, the first four bits are coded as $--+$, after which the state is σ_1. The next eight bits are coded into a six symbol word, $-+-+-+$.

TABLE II—MS43 CODE BOOK

Binary Equivalent	R_1	R_2	R_3
0000	+++	−+−	−+−
0001	++0	00−	00−
0010	+0+	0−0	0−0
0100	0++	−00	−00
1000	+−+	+−+	−−−
0011	0−+	0−+	0−+
0101	−0+	−0+	−0+
1001	00+	00+	−−0
1010	0+0	0+0	−0−
1100	+00	+00	0−−
0110	−+0	−+0	−+0
1110	+−0	+−0	+−0
1101	+0−	+0−	+0−
1011	0+−	0+−	0+−
0111	−++	−++	−−+
1111	++−	+−−	+−−

VII. CONCLUSION

This paper has presented a description of a number of properties of multilevel coding for synchronous baseband transmission. The dominant feature of such coding is the set of pulse sequence requirements imposed by channel characteristics and system operation. It was shown that multialphabet and variable length codes are a natural consequence of attempting to attain high efficiency with codes of this type.

VIII. ACKNOWLEDGMENTS

The author is indebted to Miss N. K. Shellenberger for programming support, and to J. M. Sipress for many valuable discussions.

APPENDIX

A Search Routine for the Principal States of a Ternary Sequence

This appendix describes a search routine to find the principal states and associated code words from a given ensemble of ternary sequences. The codes are of fixed word length N and the symbol sequence is required to have the following properties:

(*i*) The maximum number of different levels which can be assumed by the running digital sum (with positive, zero and negative pulses assigned values of $+1$, 0, and -1, respectively) is $2Q + 1$.

(*ii*) The maximum number of consecutive zeros is R.

TABLE III—VL43 CODE BOOK

Binary Equivalent	Alphabet for Coder State $W^*(\sigma_1)$	Alphabet for Coder State $W^*(\sigma_2)$	Alphabet for Coder State $W^*(\sigma_3)$
0000	+−+	+−+	
0001		−+−	−+−
0010	0+0	0+0	−0−
0100	+00	+−−	+−−
1000	00+	00+	−−0
0011	0+−	0+−	0+−
0101	+0−	+0−	+0−
1001	++0	00−	00−
1010	+0+	0−0	0−0
1100	0++	−−+	−−+
0110	+−0	+−0	+−0
1110	−+0	−+0	−+0
1101	−0+	−0+	−0+
1011	0−+	0−+	0−+
0111	++−	++−	0−−
1111	−++	−++	−00

Binary Equivalent	$W^*(\sigma_1)$ Alphabet	Binary Equivalent	$W^*(\sigma_3)$ Alphabet
00010000	−+−+−+	00000000	−−−+−+
00010001	+++−+−	00000001	+−+−+−
00010010	−+−0+0	00000010	+−+−0−
00010100	−+−+00	00000100	000+−−
00011000	−+−00+	00001000	+−+−−0
00010011	+++−0−	00000011	−−−+0+
00010101	+++−−0	00000101	−−−++0
00011001	−+−++0	00001001	+−+00−
00011010	−+−+0+	00001010	+−+0−0
00011100	−+−0++	00001100	+−+−−+
00010110	000++−	00000110	000+−0
00011110	000−+0	00001110	000−−+
00011101	+++−−+	00001101	−−−++−
00011011	+++−00	00001011	−−−+00
00010111	−+−++−	00000111	+−+0−−
00011111	000−++	00001111	+−+−00

(*iii*) The maximum number of consecutive pulses of like sign is $P \leqq 2Q$.

The first step is a search for the set $\{w\}$ of words of length N which satisfy the above conditions.

Let $w_j = a_{1j}a_{2j} \cdots 2_{Nj}$, with $a_{ij} = -1, 0$ or $+1$. It is then required that

$$-Q \leqq \sum_{i=p}^{r} a_{ij} \leqq +Q$$

for all r, ρ such that

$$1 \leqq r \leqq N$$

$$1 \leqq \rho \leqq r,$$

where N is the fixed word length. Moreover, if

$$a_{i,j} = a_{i+1,j} = \cdots = a_{i+P-1,j} = \pm 1$$

then

$$a_{i+P,j} \neq a_{i,j} .$$

Also, if

$$a_{i,j} = a_{i+1,j} = a_{i+R-1,j} = 0$$

then

$$a_{i+R,j} \neq 0.$$

Next, a set of allowable states $s(t,u,v)$ is defined, where t denotes the value of the running digital sum of the pulse sequence, u the preceding number of zeroes and $v > 0$ ($v < 0$) the number of preceding positive (negative) pulses. It is required that

$$-Q \leqq t \leqq Q, \qquad 0 \leqq u \leqq R, \qquad -P \leqq v \leqq P.$$

The words $\{w\}$, when used in a particular state $s(t,u,v)$ take the sequence through N states. A word w_j is permissible in $s(t,u,v)$ if and only if the N resulting states are allowable. Each word in $\{w\}$ is tested to determine the states in which it is permissible. The result is that a set of words $\{w\}_{t,u,v}$ is associated with each state $s(t,u,v)$.

When the above procedure is completed, the program searches for the principal states. States which contain less than 2^α words are discarded. Then, for each remaining state $s(t,u,v)$, those words in $\{w\}_{t,u,v}$ are eliminated which terminate in a state which has been discarded. If $\{w\}_{t,u,v}$ contains fewer than 2^α words after this operation, $s(t,u,v)$ is eliminated. This procedure is continued until either no states remain or the routine runs through a complete cycle of states without eliminating any words or states. In the latter case, the remaining states are the principal states, $\{\sigma_q\}$, and the remaining words are $\{W(\sigma_q)\}$.

REFERENCES

1. Sipress, J. M., A New Class of Selected Ternary Pulse Transmission Plans for Digital Transmission Lines, IEEE Trans. Commun. Tech., *COM-13,* No. 3, September, 1965, pp. 366–372.
2. Davis, C. G., An Experimental Pulse Code Modulation System for Short-Haul Trunks, B.S.T.J., *41,* January, 1962, pp. 1–24.
3. Aaron, M. R., PCM Transmission in the Exchange Plant, B.S.T.J., *41,* January, 1962, pp. 99–142.
4. Mayo, J. S., A Bipolar Repeater for Pulse Code Modulation Signals, B.S.T.J., *41,* January, 1962, pp. 25–97.
5. Dorros, I., Sipress, J. M., and Waldhauer, F. D., An Experimental 224 Mb/s Digital Repeatered Line, B.S.T.J., *45,* No. 7, September, 1966.